In the claims:

For the Examiner's convenience, all pending claims are presented below with changes shown in accordance with the mandatory amendment format.

1. (Currently Amended) A method, comprising:

analyzing code ~~associated with~~ using bitwise constant propagation by a language implementation system ~~using bitwise constant propagation~~;

determining, based on the analyzed code, whether an operation therein on a larger data type may be replaced [[by]] with an operation on a smaller data type having a reduced precision~~, wherein the operation on the larger data type is contained in the code~~; and

replacing the operation on the larger data type [[by]] with the operation on the smaller data type [[as]] if so determined.

2. (Currently Amended) The method of claim 1, further comprising:

determining whether a first variable of the larger data type may be replaced [[by]] with a second variable of [[a]] the smaller data type having the reduced precision; and

replacing the first variable of the larger data type [[by]] with the second variable of the smaller data type [[as]] if so determined.

3. (Original) The method of claim 2, wherein replacing the operation and replacing the first variable are used for automatic vectorization for signal and media processors that provide vector operations on small fixed-point data types.

4. (Previously Presented) The method of claim 3, wherein the processors are equipped to provide MMX instructions.

5. (Previously Presented) The method of claim 3, wherein the processors are equipped to provide SSE instructions.

6. (Currently Amended) The method of claim 3, further comprising performing algebraic simplification [[on]] of the code.

7. (Currently Amended) The method of claim 6, wherein the language implementation system performs the bitwise constant propagation by abstract interpretation [[on]] of the code.

8. (Currently Amended) A computer-readable medium having stored thereon a plurality of instructions, said plurality of instructions when executed by a computer, cause said computer to perform:

analyzing code ~~associated with~~ using bitwise constant propagation by a language implementation system ~~using bitwise constant propagation~~;

determining, based on the analyzed code, whether an operation therein on a larger data type may be replaced [[by]] with an operation on a smaller data type having a reduced precision~~, wherein the operation on the larger data type is contained in the code~~; and

replacing the operation on the larger data type [[by]] with the operation on the smaller data type [[as]] if so determined.

9. (Currently Amended) The computer-readable medium of claim 8 having stored thereon additional instructions, said additional instructions when executed by a computer for optimizing, cause said computer to further perform:

determining whether a first variable of the larger data type may be replaced [[by]] with a second variable of [[a]] the smaller data type having the reduced precision; and

replacing the first variable of the larger data type [[by]] with the second variable of the smaller data type [[as]] if so determined.

10. (Original) The computer-readable medium of claim 9 wherein replacing the operation and replacing the first variable are used for automatic vectorization for signal and media processors that provide vector operations on small fixed-point data types.

11. (Previously Presented) The computer-readable medium of claim 10, wherein the processors are equipped to provide MMX instructions.

12. (Previously Presented) The computer-readable medium of claim 10, wherein the processors are equipped to provide SSE instructions.

13. (Currently Amended) The computer-readable medium of claim 10, having stored thereon additional instructions, said additional instructions when executed by a computer for optimizing, cause said computer to further perform performing algebraic simplification [[on]] of the code.

14. (Currently Amended) The computer-readable medium of claim 13, wherein the language implementation system performs the bitwise constant propagation by abstract interpretation [[on]] of the code.

15. (Currently Amended) A system, comprising:

a processor;

memory connected to the processor ~~storing~~ to store instructions for type demotion of expressions and variables by bitwise constant propagation, the instructions to be executed by the processor;

storage connected to the processor ~~that stores~~ to store a software code having a plurality of separately compilable routines,

wherein the processor to execute[[s]] the instructions on the code to:

analyze ~~analyzing~~ code ~~associated with~~ using bitwise constant propagation by a language implementation system ~~using bitwise constant propagation~~;

determine, based on the analyzed code, whether an operation on a larger data type may be replaced [[by]] with an operation therein on a smaller data type having a reduced precision~~, wherein the operation on the larger data type is contained in the code~~; and

replace the operation on the larger data type [[by]] with the operation on the smaller data type [[as]] if so determined.

16. (Currently Amended) The system of claim 15, wherein the processor further to:

determine[[s]] whether a first variable of the larger data type may be replaced [[by]] with a second variable of [[a]] the smaller data type having the reduced precision; and

to replace[[s]] the first variable of the larger data type [[by]] with the second variable of the smaller data type [[as]] if so determined.

17. (Original) The system of claim 16, wherein the processor replaces the operation and replaces the first variable to provide vector operations on small fixed-point data types.

18. (Previously Presented) The system of claim 17, wherein the processor is equipped to provide MMX instructions.

19. (Previously Presented) The system of claim 17, wherein the processor is equipped to provide SSE instructions.

20. (Currently Amended) The system of claim 18, wherein the processor performs algebraic simplification [[on]] of the code.

21. (Currently Amended) The system of claim 19, wherein the language implementation system performs the bitwise constant propagation by abstract interpretation [[on]] of the code.

22. (Currently Amended) A system, comprising:

means for analyzing code ~~associated with~~ using bitwise constant propagation by a language implementation system ~~using bitwise constant propagation~~;

means for determining, based on the analyzed code, whether an operation therein on a larger data type may be replaced [[by]] with an operation on a smaller data type having a reduced precision~~, wherein the operation on the larger data type is contained in the code~~; and

means for replacing the operation on the larger data type [[by]] with the operation on the smaller data type [[as]] if so determined.

23. (Currently Amended) The system of claim 22, further comprising:

means for determining whether a first variable of the larger data type may be replaced [[by]] with a second variable of [[a]] the smaller data type having the reduced precision; and

means for replacing the first variable of the larger data type [[by]] with the second variable of the smaller data type [[as]] if so determined.

24. (Original) The system of claim 23, wherein the means for replacing the operation and the means for replacing the first variable are used for automatic vectorization for signal and media processors that provide vector operations on small fixed-point data types.

25. (Previously Presented) The system of claim 24, wherein the processors are equipped to provide MMX instructions.

26. (Previously Presented) The system of claim 24, wherein the processors are equipped to provide SSE instructions.

27. (Currently Amended) The system of claim 24, further comprising means for performing algebraic simplification [[on]] of the code.

28. (Currently Amended) The system of claim 27, wherein the language implementation system performs the bitwise constant propagation by abstract interpretation [[on]] of the code.